

REMARKS

In response to the Office Action mailed on April 18, 2007, Applicant(s) respectfully request(s) reconsideration.

Claims 1-3, 5-15, 17-24, 26-29 and 31-33 are now pending in this Application.

In this Amendment, claims 1, 14, 17-18, 28 and 31-33 have been amended.

Claims 1, 14, 17, 18, 28 and 31-33 are independent claims and the remaining claims are dependent claims.

Applicant(s) believe that the claim(s) as presented are in condition for allowance. A notice to this affect is respectfully requested.

Claims 1-3, 5-11, 13-15, 17-24, 27-29 and 31-33 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,085,030 (Whitehead '030). Applicants respectfully disagree with these contentions and assert that the present claimed invention is not anticipated by any disclosure in the Whitehead '030 references.

The claimed API objects in the present case are not anticipated by the Whitehead remote objects because the API objects are a limited reflection of the base objects from which they derive. Claim 1 has been herein amended to recite explicitly instantiating an API object reflective of the identified native object in a client object space, to more clearly clarify the dichotomous nature of the API object and the native object to which it corresponds. As alluded to by the previous office action, claim 1 now clearly recites specific instantiation of the API object that is reflective, but not duplicative, of the base object, discussed further at page 12, lines 11-18.

Whitehead teaches, and the Office Action agrees, returning an object "that resembles a native component" (col. 8, line 43). The previous paragraphs in Whitehead '030 make clear that Whitehead employs an object factory for generating multiple objects that are the same as a base object from which they derive. As is known to those of skill in the art, a CORBA (Common Object Request Broker) request returns an object that conforms to a native or base class. There is no difference between the base class and the instantiated objects. In contrast, the claimed API object is merely reflective of, not duplicative of, the base object to which it corresponds, disclosed further at page 16, lines 22-29.

Further, changes to the API object refer back to the native object such that a change or modification can be made in only one place (in the native object). In contrast, a Whitehead object remains an autonomous entity exhibiting its own independent set of attributes apart from the native object (col. 11, line 55-col. 12 line 3). Local instantiation merely obtains a copy of the object. Remote instantiation obtains a proxy copy which is a duplicative copy bound to the local proxy object (12:1). The claimed API object differs because the API object invokes only API related attributes and methods, thus representing only a subset of the operations of the native (Base) object. Therefore, base application objects are selectively exposed in the remote client object space (page 12, lines 6-18). Only a subset of the objects are available via the remote access object library for API object instantiations in the remote client object space, and further, within those exposed objects, only API relevant attributes and methods are available. Selective exposure is determined by metadata, as discussed at page 16, lines 22-29. Accordingly, claim 1 has been herein amended to recite that the API object selectively expos[es] the corresponding native object. Claim 17 has been similarly amended.

In other words, the API objects represent a specialized subset of the base application objects (library) by including a reflective object for API related operations, and within those exposed reflective objects, defining a further subset of available attributes and methods germane to the API specific tasks. By way of further distinction, claim 14 has been further amended to recite identifying, for each of the object types, available methods for each of the exposed objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object in the base application object library, disclosed at page 15, lines 16-25, to clarify that the API objects include a selected subset of operations and methods from the corresponding base object. Claims 18, 28 and 31-33 have been similarly amended. Further, in an overabundance of caution, claims 28 and 31 have been further amended as claim 1 above. Claims 14, 18, 28 and 31-33 are therefore respectfully submitted as allowable.

As the remaining claims depend, either directly or indirectly, from claims 1, 14, 18 and 28, it is respectfully submitted that all claims are now in condition for allowance.

-16-

Applicant(s) hereby petition(s) for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-3735.

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned collect at (508) 616-9660, in Westborough, Massachusetts.

Respectfully submitted,

/CJL/

---

Christopher J. Lutz, Esq.  
Attorney for Applicant(s)  
Registration No.: 44,883  
Chapin Intellectual Property Law, LLC  
Westborough Office Park  
1700 West Park Drive  
Westborough, Massachusetts 01581  
Telephone: (508) 616-9660  
Facsimile: (508) 616-9661

Attorney Docket No.: EMC03-18(03088)

Dated: July 17, 2007